LiveMicro: An Edge Computing System for Collaborative Telepathology

LiveMicro: An Edge Computing System for Collaborative Telepathology

Alessio Sacco*
Politecnico di Torino

Flavio Esposito
Saint Louis University

Princewill Okorie
Saint Louis University

Guido Marchetto
Politecnico di Torino

Abstract

Telepathology is the practice of digitizing histological images for transmission along telecommunication pathways for diagnosis, consultation or continuing medical education. Existing telepathology solutions are limited to offline or delay-tolerant diagnosis.

In this paper we present *LiveMicro*, a telepathology system that, leveraging edge computing, enables multiple pathologists to collaborate on a diagnosis by allowing a remote live control of a microscope. In such environment, computation at the edge is used in three ways: (1) to allow remote users to control the microscope simultaneously, (2) to process histological image and live video, by running algorithms that recognize *e.g.*, tumor grades, (3) to preserve privacy creating virtual shared data views. In particular, we built the first open-source edge computing based telepathology system. In our prototype, the examples of edge processing that we currently support are extraction of diagnosis-oriented features and compression of payloads to minimize transmission delays. Our evaluation shows how LiveMicro can help a medical team with a remote, faster and more accurate diagnosis.

1 Introduction

Pathologists nowadays diagnose histological images with a physical multi-lens microscope, manually moving glass-slides, adjusting focus and switching lens to explore the area of interest. Often pathologist analyzes histological images on a glass slide while the patient is still under a tumor removal surgery. Hence, a quick pathology assessment is crucial for the patient. In the vast majority of non-trivial pathology cases, to minimize the time to response to the surgeon team and the probability of incorrect assessments, pathologists ask for second opinions to nearby experts (if available) by physically carrying privacy protected glass specimens. Hospitals in rural areas are often forced to outsource pathology cases, incurring significant delays. Telepathology can be used to connect experts with patient data, allowing transmission of high-resolution images of specimens. This allows a rapid diagnosis of complex cases even in geographical areas that lack local expertise.

Current telepathology solutions are limited by the technology, the (best-effort) performance of the underlying telecommunication media on which they rely on, *i.e.*, the Internet. At best, they use a virtual private network for in-hospital offline, *i.e.*, non-real-time, consultations. Moreover, they are not open-source (see § 2). Telepathology today is practically unused for the applications that would need it the most: (i) delay and bandwidth sensitive data processing and sharing, (ii) fast and reliable remote consultations, and (iii) multi-students live teaching sessions.

Our contribution. To this end, we propose *LiveMicro*, an edge computing based telepathology system whose goal is to allow real-time remote control of the microscope and real-time histological image processing. In particular, we empower an open-source microscope firmware [15] with capabilities of rapid processing of histological images, and with low-latency image transmissions. Examples of edge computing tasks that we validate include (i) payload optimization methods such as compression algorithms and high resolution image format conversion, and (ii) pattern matching algorithms to identify biological markers of tumors on the image under investigation and suggest automatically the tumor grade.

LiveMicro is composed of several components (§ 3). The web based front-end allows pathologists located in remote locations to sign up, login, join a session and interact with a microscope. By interaction we mean pan, zoom, or capture images for live or subsequent processing. Commands are sent via our own protocol (based on gRPC [10]) so that feedback on the remote microscope action is immediate when enough network bandwidth is available. The back-end of LiveMicro is instead composed of an edge cloud based on an enhanced version of OpenStack. Images or videos are captured by the microscope, can be pre-processed at network edge and then are sent, digitally, to the pathologist client via a web server. Furthermore, we modify the original OpenStack Queens to

*The work of A.Sacco was performed while at Saint Louis University.

integrate the edge network guaranteeing adequate scheduling of the resources.

To validate our system and highlight the edge computing advantages, we analyzed its performance in different use cases, using a microscope emulator and a real prototype(§ 4). Our results are promising and demonstrate how edge computing environments could tremendously help the field of pathology.

2 Related Work

Telepathology. All existing telepathology systems are based on time-consuming digital compositions and transmission of large images captured from cameras attached to the microscope [24]. To our knowledge, the oldest attempt to remotely control a motorized video-microscope was in 1991 [22]. Notably, the researchers had to reserve enough bandwidth from Norway Telecommunication to transfer their images. Nowadays, advanced layer2 network functionally used by GENI, Internet2 [20], and ESnet [19] can extend those high-performance paths across the regional, national, and international network transit providers. More recent solutions have attempted to connect a microscope over virtual paths with guaranteed performance. R. Weinberg for example [5] used the GENI testbed [2] to stream videos (from Los Angeles to Chattanooga) captured from a remotely located microscope for high school biology education. Other studies designed expensive arrays of microscope processors to capture images [23], as well as inexpensive solutions using images captured using smartphones [7] or cameras on board of a Raspberry Pi for telecytology [6]. In the latter solution, images were transferred using FaceTime (hence in a broadcast, without the ability to remotely control the microscope).

We share with such solutions the affordability and the high-throughput goals. But in addition, we support low latency access to a remotely controlled microscope [1, 8] and the ability to process imagery at the edge of the network before or after their transfer.

Edge Computing for Image and Video Processing. Many solutions have been proposed with the aim of processing images and videos at the edge. Some focusing on the back-end infrastructure [3, 12, 18], some focusing on the mobile edge computing paradigm [4, 13, 14]. In all these cases, as in LiveMicro, detection requests for patterns or objects are been processed in proximity of the source of information.

Despite being intriguing and based on novel and sound approaches, these solutions differ from ours as they lack a collaborative cyber-human interaction. Our goal has been to design and implement a system that would allow remote control of human pathology gestures on a microscope, via a "software-defined glass slides". This is fundamentally different from a video conference, *e.g.*, for a remote surgery application. In our telepathology system, the resolution, responsiveness, and size of the histological images to transfer and preprocess can be dauntingly large but very low delays and high throughput are required. None of these requirements were simultaneously tackled in previous solutions.

3 LiveMicro: Architecture and Processing

In this section we describe the design and implementation details of LiveMicro. Our designed has focused on allowing (i) *remote computations* and (ii) *remote consultations*. In spite of pathology applications, we argue that any field that uses a microscope, for example microbiology, may benefit from our system. By remote consultation we mean the possibility for pathologists to request (web) access through our front-end interface to a live session of a microscope and remotely control its firmware and its functionalities. Typically pathologists ask for panning, zooming or taking snapshots of histological samples. To manage each telepathology (LiveMicro) session, our back-end OpenStack driven edge cloud assigns a dedicated Virtual Machine (VM) to each user. We leave as an open problem the performance comparison of our VM-based solution with other virtualization technologies, such as Linux Containers or unikernels.

3.1 Front-End and Plugin Design

As shown in Figure 1, our design goal is for pathologists to be able to access the microscope in a user-friendly manner, by merely using a web browser. Our web server is the entry point for the entire system and acts as a portal through which users connect to the LiveMicro ecosystem and start, join, or terminate one or multiple telepathology sessions. Our front-end design goal was to be as lightweight as possible, but with an intuitive and user-friendly design. Our web interface is implemented in AngularJS.

At the other end of the telepathology session, a microscope runs a modified version of *μManager* — often named Micro-Manager, as in microscope-manager — an open-source package for controlling and configuring a fairly large amount of commonly used microscopes. *μManager* did not support network connectivity nor edge computing functionalities. Our modified instance of *μManager* can be plugged to a physical machine attached to a microscope, to handle data marshaling between the network and the microscope firmware. It can also be attached to a microscope emulator. Our prototype uses an Olympus IX81 [16], one of the microscopes whose interface is compatible with Micro-Manager. Livemicro also supports an emulated version of the microscope, which may be a very effective tool to scale, for example for pathology medical education.

Since pathologists (in training or at work) need to switch microscope lens, and examine tissues looking for patterns, a simple image snapshot is often not enough. Live streaming of the sample under consideration is hence necessary to have an immediate feedback and make the system usable. We use

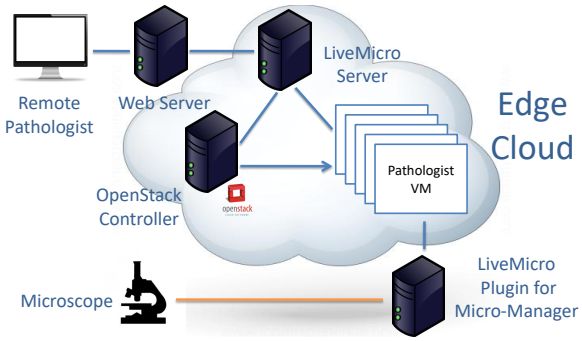

Figure 1: LiveMicro services are spread across the infrastructure, the microscope uses a dedicated machine and a dedicated hardware for capturing samples.

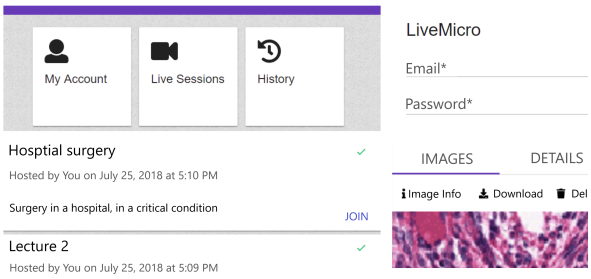

Figure 2: Screenshots of the web interface of our application. From the top left to the bottom left it shows: home page, login page, real-time view of histological images, list of live sessions currently active.

ffmpeg [9] to encode and transmit videos between our Micro-Manager Plugin (from now on denoted as, Plugin) and the LivMicro Server, while on the web-page, our *WebRTC* [21] plugin is responsible for receiving and showing the video. Our edge cloud pre-processes the stream compressing its payload. The compression is not performed on the Plugin, but in a second phase, thereby videos can be stored for pre-processing or retrieved at a later stage.

3.2 Core and Edge Cloud Management

To control large pools of compute, storage, and networking resources between the web server and the LiveMicro μManager plugin, we deployed our own Edge Cloud infrastructure (Figure 1), modifying OpenStack, a well-known open-source Cloud Computing platform. We associate each user of a telepathology session to a VM; if needed, each VM can provide network or node functionalities, *e.g.*, CPU-intensive algorithms on the histological imagery.

Our edge computing architecture requires at least two nodes (hosts) responsible for launching the core management functionalities: the *controller node* and the *compute node*. The controller manages the resources available in the infrastructure

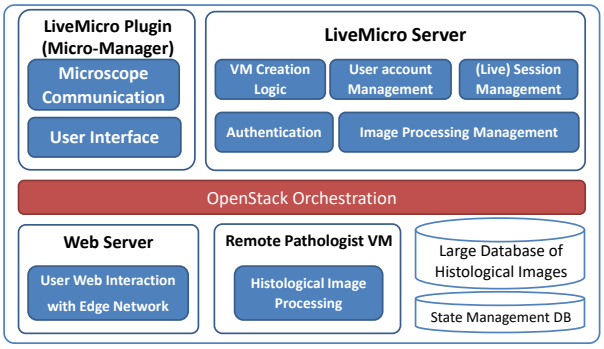

Figure 3: Overall LiveMicro architecture: blue box are our implemented services (everyone except the OpenStack orchestration).

and the compute node runs the VMs and their bookkeeping; the networking service agent then connects all telepathology instances to their isolated virtual networks providing firewalling services to instances via security groups.

The choice of the best hosting machine is based on some configurable policies, *e.g.*, hosted application requirements or the usage of machines at that moment. By default, OpenStack selects the hosting machine independently from the application logic. But in our scenario we forced the controller node to choose a node near the requested microscope, to guarantee low delay. Similarly to OpenStack++ [11], we modified the default cloud orchestration mechanisms to better support edge computing applications. Differently from OpenStack++, however, we focus on modifying the VM scheduler to better support multiple telepathology sessions.

3.3 LiveMicro Server

In Figure 3 we present the architecture of LiveMicro, highlighting the key mechanisms provided by each component. The LiveMicro Server is the core process that runs most of the logic of our application. It can receive requests from the web server or from the microscope (emulator) plugin, and it is in charge of deciding the operations to perform: it communicates with the database, it manages live sessions and decides when it is time to create or destroy a VM, according to the business logic.

The LiveMicro Server also provides the services required to remotely control the microscope. In addition, this server is responsible for the management of the prior (live) sessions and to handle the Node.js API REST calls launched by the desktop or the emulator plugin application.

A telepathology session client runs on each VM and acts as a proxy: it receives requests from LiveMicro Server, elaborates them and in case sends them to the plugin. The requests are not forwarded if the response is already in process on the VM, *e.g.*, the client is working on an already snapped image.

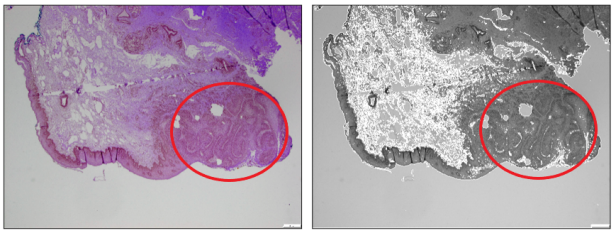

Figure 4: Currently, pathologists manually count expressed cells. The count of cells that co-express (Olig-2 and Ki-67 histological markers) is used to assess tumor proliferation and hence can change the course of an active surgery or a treatment. (Left) View of invasive squamous cell carcinoma of a tongue, with tumor area highlighted in the red circle. (Right) Image after the application of nuclei detection algorithm: in addition to the image, information about percentage of marker co-expression in the sample is automatically provided to the web interface.

This cache layer is necessary to avoid overloading the real microscope.

The (image or video) processing of each telepathology session can span across multiple VMs. It is hence possible for each client to ask different processing, at the same time, on the same virtualized histological image. The framework guarantees that actions of a user do not affect the analysis of another pathologist working on the same sample, because the operations take place in different VMs.

Typical image processing operations performed on the samples can be: color decomposition, count of nuclei of proliferating tumor cells, detection of pattern tumor as the user moves the image, etc. Figure 4 shows an example of image processing implemented in ImageJ and OpenCV; in particular, the counting of affected nuclei. The counting is recomputed every time the image is zoomed or moved. Knowing the percentage of cells activated by different markers (i.e., colors) is used by the pathologist to assess the tumor grade (first stage, advanced stage, etc.)

4 Evaluation Results

Evaluation scenario and testbed. To establish the practicality of our design, we developed a testbed that was deployed both on CloudLab [17] and on our own servers. To test our edge computing-based telepathology system, we used the emulated version of the microscope as described in Section 3. We installed all services across three physical machines while the VMs were deployed by the OpenStack orchestrator.

In a typical scenario, the pathologist asks for a service, *e.g.*, to join a session and remotely control the microscope. To cope with the lack of a real end-user in the emulated version, and to test our system performance, we replaced the front end web server with a request generator. The requests are sent directly to the LiveMicro server, which receives responses and evaluates the encountered end-to-end latency within our edge cloud. The LiveMicro Server also multiplexes and demultiplexes network requests to/from the proper VMs.

Image and Video Performance Analysis. We tried to assess the main benefits of using edge computing in a telepathology session by measuring delays when using additional computational capacity at the edge, unavailable on the microscope. Our validated hypothesis was that such computations could, in turn, help a team of pathologists with their diagnosis and speed up the image transfer from the microscope. To this aim, we process a set of images and quantify the often expected system performance improvement. We run our microscope emulator on the local testbed, while the servers are hosted on Cloudlab bare metal machines.

We begin by quantifying the time required for a pathologist to receive, compress and process an image. The image processing instead entails the nuclei count, for example to assess the tumor grade and whether or not it is a tumor. Tumor cells under analysis react to different histological markers after bio-reagents such as Ki-97 have been added.

We compare image elaboration requests by a pathologist under three use cases (Figure 5): local image processing (Local), edge processing (Edge) and core cloud processing (Cloud). When using the *Edge* use case, the image elaboration is performed at the edge of the network, by a machine in local proximity of the microscope. After the processing occurs the result is then compressed and sent back for use. In our edge experiment we process on host machines running Ubuntu, Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz, while the VMs are limited to 1VCPU, 2GB RAM and 20GB Disk.

Similarly, in the *Cloud* use case, the images are first preprocessed by a server in the cloud, and then sent compressed to the client. In this case the latency between the microscope and the server has a larger impact. Host machines are Ubuntu, Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz, while the VMs have the same constraints of the previous case.

Conversely, in the *Local* case, the original image is sent to the client, to run the image processing algorithm on it. In this scenario, the image sent needs to be uncompressed. This is because the calculation is better performed on the original version where the pixel information is maintained as close to the original as possible. On the contrary, an elaboration on a compressed image can lead to erroneous diagnosis. Note that host machines are Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz.

Figure 5a reflects our considerations, and shows the processing time for each use case: *Edge* reduces the latency by as many as 25% with respect to a *Cloud* processing and markedly by more than 30% with respect to the *Local* processing solution.

With our available hardware, we found that with the *Edge* we can reduce the elaboration time w.r.t. the *Local* case con-

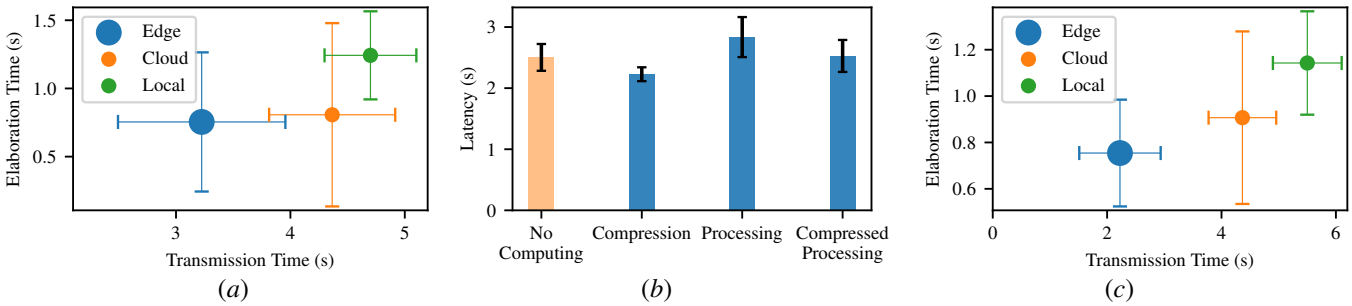

Figure 5: Edge Computing Advantages: (a) Elaboration and Transmission time of image processing performed locally, on the cloud and on the edge. (b) Time necessary for different operations to be performed at the edge plus image transmission time. (c) Elaboration and Transmission time per frame of video processing performed locally, on the cloud and on the edge. For streaming video results show how edge improve the transmission, even better than images transmission. All bars have 95% C.I.

sidering that a more powerful machine computes processing. Moreover, transmission time is less than *Cloud* case, because computation is closer to the source. Likewise, Figure 5b quantifies our latency tests when processing image samples of different sizes. Values are obtained averaging cases in which the image is cached on the server (typical scenario) and the image is retrieved from the database (worst-case scenario). Four situations are taken into account: (*i*) *No computing*, non-processed image is sent, (*ii*) *Compressed image*, no processing on the sample is performed but the image is sent compressed, (*iii*) *Histological processing*, one image elaboration algorithm (counting nuclei) is applied on the sample sent without compression, (*iv*) *Compressed histological processing*, one image elaboration algorithm (counting nuclei) is applied on the sample sent after compression. The latency for the latter use case is comparable to the one with no edge computing, confirming how the processing at the edge leads to tangible benefits.

In addition to the image transmission, we tested the live video streaming use case. In Figure 5c we compared the advantages of edge computing for real-time video transmission. The latency shown in the graph represents the lag between the release of a new frame by the microscope and the instant after which it has been received by the web client. Similarly to Figure 5a we tested the 3 use cases aforementioned and in the same testbed. However, for video, we obtain a considerable 46% improvement using edge over cloud. This value is even higher than the result obtained for image transmission.

It is obvious that compression reduces latency, in particular, it reduces transmission time given the smaller compressed payload. However, this section demonstrated the need and use of even a simple pre-processing at the edge to handle a telepathology session. This study is the first, to our knowledge, to have merged these two technologies, telepathology and edge computing. The two fields have been singularly and extensively studied before, but in conjunction may help save lives through faster and more accurate diagnosis via remote live consultations.

5 Conclusion

Telepathology, the practice of pathology at long distance by pathologists has been around since 1986 but never took off due to poor performance and the lack of usability. In this paper, we presented LiveMicro, a system that has the potential to advance significantly the field of telepathology by augmenting live remote microscope session with the computational power of (cutting) edge computing technologies. LiveMicro allows a team of pathologists to access, control and process, simultaneously, a remotely located (real or emulated) microscope using merely a (mobile) web browser. We presented the architecture of our prototype and disclosed its potentials to improve the field of medical diagnosis in critical situations, for example under an active surgery, where a quick diagnosis is literally vital but experts are locally unavailable.

We demonstrated with some initial results how an edge computing-empowered microscope may provide additional benefits such as speeding up image and video transmission time and performing application-specific image processing. Such processing involved tumoral cell identification to speed-up pathology diagnosis (currently expressed cells are manually counted) and to support pathologies in continuous education.

Acknowledgments

This work has been partially supported by Saint Louis University Presidential Research Fund. We would like to thank Dr. Grant Kolar, M.D. and Dr. Katherine Schwetye, M.D., Ph.D., for their help and guidance on the pathology aspects of this project.

Availability

A demonstration video of our system can be seen at https://live-micro.gitlab.io.

References

[1] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh. Realizing the tactile internet: Haptic communications over next generation 5g cellular networks. *IEEE Wireless Communications*, 24(2):82–89, April 2017.

[2] Mark Berman, Chip Elliott, and Lawrence Landweber. Geni: Large-scale distributed infrastructure for networking and distributed systems research. In *Communications and Electronics (ICCE), 2014 IEEE Fifth International Conference on*, pages 156–161. IEEE, 2014.

[3] Dmitrii Chemodanov, Flavio Esposito, Andrei Sukhov, Prasad Calyam, Huy Trinh, and Zakariya Oraibi. Agra: Ai-augmented geographic routing approach for iot-based incident-supporting applications. *Future Generation Computer Systems*, 92:1051–1065, 2019.

[4] Xu Chen, Lei Jiao, Wenzhong Li, and Xiaoming Fu. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 24(5):2795–2808, 2016.

[5] Digital Tele-Microscopy in Support of Teaching Biology. http://grantome.com/grant/NSF/CNS-1451220.

[6] Radu Dudas, Christopher VandenBussche, Alex Baras, Syed Z Ali, and Matthew T Olson. Inexpensive telecytology solutions that use the raspberry pi and the iphone. *Journal of the American Society of Cytopathology*, 3(1):49–55, 2014.

[7] Donald U Ekong and Paul Fontelo. Prototype telepathology solutions that use the raspberry pi and mobile devices. In *Global Humanitarian Technology Conference (GHTC), 2017 IEEE*, pages 1–4. IEEE, 2017.

[8] Gerhard P Fettweis. The tactile internet: Applications and challenges. *IEEE Vehicular Technology Magazine*, 9(1):64–70, 2014.

[9] FFmpeg online documentation. https://www.ffmpeg.org/.

[10] gRPC, A high performance, open-source universal RPC framework. https://grpc.io/docs/.

[11] Kiryong Ha and Mahadev Satyanarayanan. Openstack++ for cloudlet deployment. *School of Computer Science Carnegie Mellon University Pittsburgh*, 2015.

[12] Changchun Long, Yang Cao, Tao Jiang, and Qian Zhang. Edge computing framework for cooperative video processing in multimedia iot systems. *IEEE Transactions on Multimedia*, 20(5):1126–1139, 2018.

[13] Pavel Mach and Zdenek Becvar. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3):1628–1656, 2017.

[14] Yuyi Mao, Jun Zhang, and Khaled B Letaief. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications*, 34(12):3590–3605, 2016.

[15] Micro-Manager Open-source Microscopy Software. https://micro-manager.org/.

[16] Olympus IX81 microscope. https://www.biocompare.com/19419-Inverted-Microscopes/399623-IX81-Inverted-Microscope/.

[17] Robert Ricci, Eric Eide, and The CloudLab Team. Introducing CloudLab: Scientific infrastructure for advancing cloud architectures and applications. *USENIX Magazine ;login:*, 39(6), December 2014.

[18] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[19] The Energy Science Project. https://www.es.net/.

[20] The Internet2 Project. https://www.internet2.edu.

[21] WebRTC free and open project. https://webrtc.org/.

[22] Ronald S Weinstein, A.K Bhattacharyya, Anna R Graham, and John R Davis. Telepathology: A ten-year progress report. *Human Pathology*, 28(1):1 – 7, 1997. http://www.sciencedirect.com/science/article/pii/S0046817797902707.

[23] Ronald S Weinstein, Michael R Descour, Chen Liang, Gail Barker, Katherine M Scott, Lynne Richter, Elizabeth A Krupinski, Achyut K Bhattacharyya, John R Davis, Anna R Graham, et al. An array microscope for ultrarapid virtual slide processing and telepathology. design, fabrication, and validation study. *Human pathology*, 35(11):1303–1314, 2004.

[24] Ronald S Weinstein et al. Telepathology overview: from concept to implementation. *Human pathology*, 32(12):1283–1299, 2001.